AMENDMENTS

In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("—") being deleted:

1. (Currently Amended) A method of reducing the number of texture cache cycles while performing anisotropic mip-mapping, comprising:

mapping a target pixel needing texture to one or more texels in a higher resolution texture array, a region of support in the higher resolution texture array being defined by a long and a short axis and being generally elliptical and a level of detail being derived only from the short axis; and

performing a filtering function along an axis using the texels from the higher resolution texture array to simulate a filtering effect of using texels from the higher resolution texture array and a second texel array having a lower resolution, wherein only one level of texture level is stored and used to generate the lower resolution texture array, ~~and~~ wherein simulating the filtering effect reduces the number of cache memory cycles associated with the filtering function<u>, wherein the step of performing a filtering function further comprises:</u>

<u>using the texels from the higher resolution texture array to derive texels of the lower resolution texture array,</u>

interpolating the texels from the higher resolution texture array to form a first blended texel,

interpolating the texels from the lower resolution texel array to form a second blended texel, wherein the sum of a first product, $m*T_{AB}$, and a second product, $(1-m) T_{CD}$ is formed, where m=1/2Uf+1/4, which indicates a coordinate position of the target pixel in the lower resolution texture array, and T_{AB} and T_{CD} are adjacent texels in the lower resolution array; and

interpolating the first blended and second blended texels to arrive at a texture for the target pixel.

2. (Cancelled)

3. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of using the texels from the higher resolution texture array to derive texels of the lower resolution texture array includes filtering adjacent texels in the higher resolution texture array, based on the mapped position of the target pixel in the higher resolution texture array, to derive texels in the lower resolution texel array.

4. (Original) A method of performing anisotropic mip-mapping, as recited in claim 3, wherein four adjacent texels in the higher resolution array are used to derive two adjacent texels in the lower resolution array.

5. (Original) A method of performing anisotropic mip-mapping, as recited in claim 4, wherein an adjacent pair of texels in the higher resolution array is filtered to provide a single texel in the lower resolution array.

6. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of using the texels from the higher resolution texture array to derive texels of the lower resolution texture array includes averaging adjacent texels in the higher resolution texture array, based on the mapped position of the target pixel in the higher resolution texture array, to derive texels in the lower resolution texel array.

7. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of interpolating the texels from the higher resolution texture array to form a first blended texel includes bilinearly interpolating adjacent texels based on the mapped position of the target pixel in the higher resolution texture array.

8. (Original) A method of performing anisotropic mip-mapping, as recited in claim 7, wherein a selected texel to which the target pixel is mapped and an adjacent texel are interpolated based on the position of the target pixel in the selected texel.

9. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of interpolating the texels from the higher

resolution texture array to form a first blended texel includes forming the sum of a first product, Uf*T_B, and a second product, (1-Uf) T_C, where Uf indicates a coordinate position of the target pixel in the higher resolution texture array, and T_B and T_C are adjacent texels in the higher resolution array.

10. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of interpolating the texels from the lower resolution texel array to form a second blended texel includes bilinearly interpolating adjacent texels based on the mapped position of the target pixel in the lower resolution texture array.

11. (Original) A method of performing anisotropic mip-mapping, as recited in claim 9, wherein a selected texel to which the target pixel is mapped and an adjacent texel are interpolated based on the position of the target pixel in the selected texel.

12. (Cancelled)

13. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of interpolating the first blended and second blended texels to arrive at a texture for the target pixel includes bilinearly interpolating the first and second blended texels, based on a parameter that indicates the level of detail between and including the texels of the higher and lower resolution arrays.

14. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the step of interpolating the first blended and second blended texels to arrive at a texture for the target pixel includes forming the sum of a first product, (Df)*the first blended texel, and the second product, (1-Df)*the second blended texel, where Df is the parameter indicating the level of detail, wherein, when Df is 0, the level of detail corresponds to the lower resolution texel array, and when Df=1, the level of detail corresponds to the higher resolution texel array.

15. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein a first number of texels are sampled in a first direction in the higher resolution array and a second number of texels are sampled in a second direction in the higher resolution array to form first blended pixels in the first direction and first blended pixels in the second direction, and to form second blended pixels in the first direction and second blended pixels in the second direction, said first number being different from said second number.

16. (Currently Amended) A method of performing anisotropic mip-mapping, as recited in claim [[2]] 1, wherein the target pixel color is blended according to the following function, target pixel color=□WSi*ColorSi, for i from 1 to the number of samples, wherein □WSi=1, for i from 1 to the number of samples, and WSi is a weighting function for the ith sample, and ColorSi is the blended color for the ith sample.

17. (Original) A method of performing anisotropic mip-mapping, as recited in claim 16, wherein the blended color ColorSi of the ith sample is blended according to the following function, ColorSi=∑Wuv*Color(u,v), for u from 0 to 3, and v from 0 to 3, for the ith sample, wherein Color(u,v) is a color corresponding to Wuv on the same mipmap level, ∑Wuv=1, for u from 0 to 3, and v from 0 to 3, and Wuv is a weight coefficient for said Color(u,v).

18. (Original) A method of performing anisotropic mip-mapping, as recited in claim 17, wherein the weight coefficients are elements of a weight coefficient array, 7 W uv = $[0.25 D_f (1 - U_f) (1 - V_f) (1 - 0.75 D_f) (1 - U_f) (1 - V_f) (1 - 0.75 D_f) (1 - U_f) (1 - V_f) (1 - 0.75 D_f) (1 - U_f) (1 - V_f) (1 - 0.75 D_f) (1 - U_f) V_f (1 - 0.75 D_f) (1 - U_f) V_f 0.25 D_f (1 - U_f) V_f (1 - 0.75 D_f) (1 - U_f) V_f (1 - 0.75 D_f) U_f (1 - V_f) 0.25 D_f U_f (1 - V_f) (1 - 0.75 D_f) U_f (1 - V_f) (1 - 0.75 D_f) U_f (1 - V_f) (1 - 0.75 D_f) U_f V_f (1 - 0.75 D_f) U_f V_f (1 - 0.75 D_f) U_f V_f 0.25 D_f U_f V_f]$, where U_f, V_f are position parameters in direction u and v, respectively, and D_f is a fraction value of the level of detail.